A Review on Effect of Load Balancing on Performance Testing in Cloud Environment

Ekta
MTech CSE
DCRUST Murthal, India

Rajvir Singh
Assistant Professor CSE
DCRUST Murthal, India

ABSTRACT

Now-a-days Cloud environment is growing very swiftly among different technologies. Thus, it impacts various areas like IT, business, information architecture, software engineering and data storage. Also, testing is as important to cloud as it is to any other field. Cloud testing refers to different terms like testing the cloud, cloud testing, and a combination of these two. Along with testing there are several challenges to cloud environment. Among them, load balancing is a major constraint for cloud environment. Load balancing is a term for various distribution techniques that helps in distribution of work among various nodes. Currently, various parameters available for measurement of load balancing algorithms like throughput, response time, overhead, fault tolerance, migration time, resource utilization, scalability and performance. In this paper a study is carried out describe all these terms along with the study of these parameters related to various load balancing algorithms.

Keywords

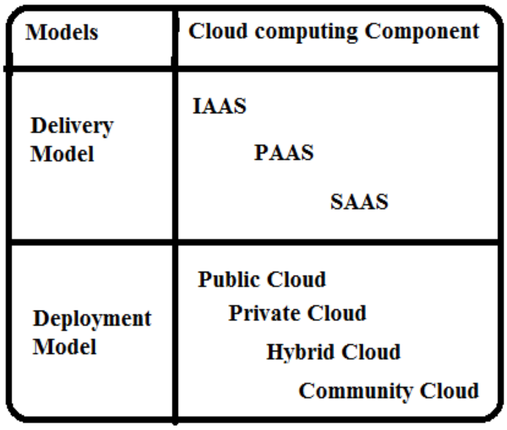
Cloud Environment, Cloud Testing, Load Balancing, Performance Testing.

1. INTRODUCTION

Cloud computing is combination of two terms, first is cloud and second is computation. First term refers to the internet that’s why sometimes cloud computing also refers as a type of internet based computing. The term cloud has been used to refer to platforms of distributed computing. Second term computation refers to the mathematical and logical methods. In general, cloud computing can be described as a combination of already existing computing technologies like distributed computing, grid computing and computing. There are various firms which are excellently working for the empowerment of cloud with time. Amazon.com is one of them and introduced the Elastic compute cloud in 2006 at very large commercial scale. Along with amazon, there are companies like Enomaly, Google, Microsoft,Netsuite, Rackspace, RightScale, salesforce.cometc.There are mainly three different types of delivery model are available. These are SaaS(Software-as-a-service), PaaS(Platform-as-a-service), IaaS(Infra-structure-as-a-service) as well as public cloud, private cloud, community cloud and hybrid cloud[1].

Major constraints or challenges for cloud environment [2][3] :

1. Security
2. Latency
3. Resource scheduling
4. Efficient load balancing
5. Performance monitoring
6. Scale and QoS management
7. Platform or language constraints
8. Consistent and robust service abstractions
9. Requires a fast speed internet connection



Models	Cloud computing Component
Delivery Model	IAAS PAAS SAAS
Deployment Model	Public Cloud Private Cloud Hybrid Cloud Community Cloud

Fig 1: Model of Cloud Computing

2. LITERATURE REVIEW

Cláudio Teixeira[1] provide information regarding PaaS(Platform as a service) which can be used as a platform to deploy their applications on cloud by minimizing the level of lock in effect.

Rajwinder kaur[2] introduced metrics for load balancing are throughput, fault tolerance, migration time, response time, scalability. Load balancing algorithms are divided into two categories: - static load balancing algorithm and dynamic load balancing algorithm:

1) Static approach: - This approach is mainly defined for the design or implementation of the system. Static load balancing algorithms split the traffic equivalently between all servers.

2) Dynamic approach: - This approach considered only the current state of the system during load balancing decisions. Dynamic approach is more suitable for widely distributed systems such as cloud computing. Dynamic load balancing approaches have two types.

They are distributed approach and non-distributed (centralized) approach. It is defined as following:

i) Centralized approach: - In centralized approach, only a single node is responsible for management and distribution of whole system. Rest nodes are not responsible.

ii) Distributed approach: - In distributed approach, each node independently constructs its own load vector. Vector gathers the load information for rest of the nodes. All decisions are made locally based on local load vectors. Distributed approach is more suitable for widely distributed systems such as cloud computing.

Soumya Ray[3] perform performance qualitative analysis on existing VM load balancing algorithm in CloudSim and java language. Execution analysis of the simulation draws a direct relation between MIPS and response time.

Zubair Khan[4] introduced how to use the general classification of different load balancing algorithms to construct the most suitable load balancing algorithms for different parallel algorithms. This classification is also considered helpful for programmer to compare different load balancing algorithms.

Shanti Swaroop Moharana[5] discuss the importance of proper implementation of load balancing algorithms in cloud environment. Overloading of a system may lead to poor performance and because of failure in any node within the network can lead to unavailability of data. So, it is important for us to study various load balancing algorithms along with their area of applicability.

Rajesh George Rajan[6] survey various existing load balancing methods in different environments and compare their parameters, merits and demerits.

R. Raju[7] discussed the job scheduling, which is a combinatorial optimization problem in the field of computer science. In this paper, a new hybrid algorithm is proposed in which a combination of ACO and cuckoo search is used in improving performance and decreasing makespan.

Subhadra Bose Shaw[8] discussed issues related to VM migration because VMs have to be moved to deal with the problem of over utilization and under-utilization of resources. In this paper, two meaning of load balancing are considered: first, it sends a large number of processes concurrently to multiple nodes to reduce turnaround time; second, it distribute the complex calculation, which causes heavy load, to multiple nodes to improve the resource utilization of each node.

J. Uma[9] discuss the primary objective of load balancing as the satisfaction of consumer by distributing the load equally among the nodes and to make maximum utilization of resources in the stipulated time. This paper compares the various load balancing algorithms and their features. These results can be used to develop an energy efficient load balancing algorithms whose focal point is less energy consumption and carbon emission factors.

Klaithem Al Nuaimi[10] discuss the various static and dynamic load balancing algorithms like INS(Index Name Server) and DDFTP(Dual Direction Downloading Algorithm from FTP servers) with fast speed and fault tolerance capabilities. Both algorithms also work in heterogeneous environment.

Martin Randles[11] present a comparative study of three distributed load balancing algorithms honey bee foraging, active clustering and random sampling walk for cloud computing. Further, study of these algorithms in context of virtualization.

Tin-Yu Wu[12] propose novel data center management architecture, which integrates de-duplication and access point selection optimization techniques to heighten the performance of cloud storage system.

Wang Jun [13] introduced that cloud computing is a way of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. In a typical application system of cloud computing, there are six layers components - client, service, application, platform, storage and infrastructure. cloud testing is a new form of software testing in which web applications use cloud computing environments to imitate real-world user traffic as a means of load testing and stress testing websites.

3. LOAD BALANCING

Load Balancing is a field within cloud computing environment which is widely elaborated with time. Various algorithm of different type are available as a solution for load balancing in cloud environment. A few major goals of load balancing algorithms can be listed as [2], [3]:

1. **Cost effectiveness:** Primary aim is to achieve an overall advancement in system performance at a reasonable cost.
2. **Scalability and flexibility:** The distributed system in which the algorithm is enforced may change in size or topology. So, the algorithm must be scalable and flexible enough for appropriate changes.
3. **Priority:** Prioritization of the resources or jobs need to be done in advance through the algorithm itself for better service, to the important or high prioritized jobs regardless of equal service provision for all the jobs regardless of their origin.

Metrics for load balancing algorithms are [2][5][6][7][14]:-

1. **Throughput: -** To calculate the all tasks whose execution has been completed.

2. **Fault Tolerance: -** Capability to recover from failure.

3. **Migration time: -** Total time to migrate the jobs or resources from one node to another.

4. **Response time: -** Amount of time that is taken by a particular load balancing algorithm to response a task in a system.

5. **Scalability: -** Ability of an algorithm to perform load balancing for any finite number of nodes of a system.

6. **Execution Time: -** Total time taken for the completion of three processes i.e. time for formation of virtual machines, processing time of a process and destruction time of a virtual machine.

7. **Makespan:-** It is also known as completion time. Makespan or completion time is the overall time taken to process a set of jobs for its complete execution.

Fig.2 represents classification load balancing algorithms. The two major categories of classification is static and dynamic load balancers [2][4]:-

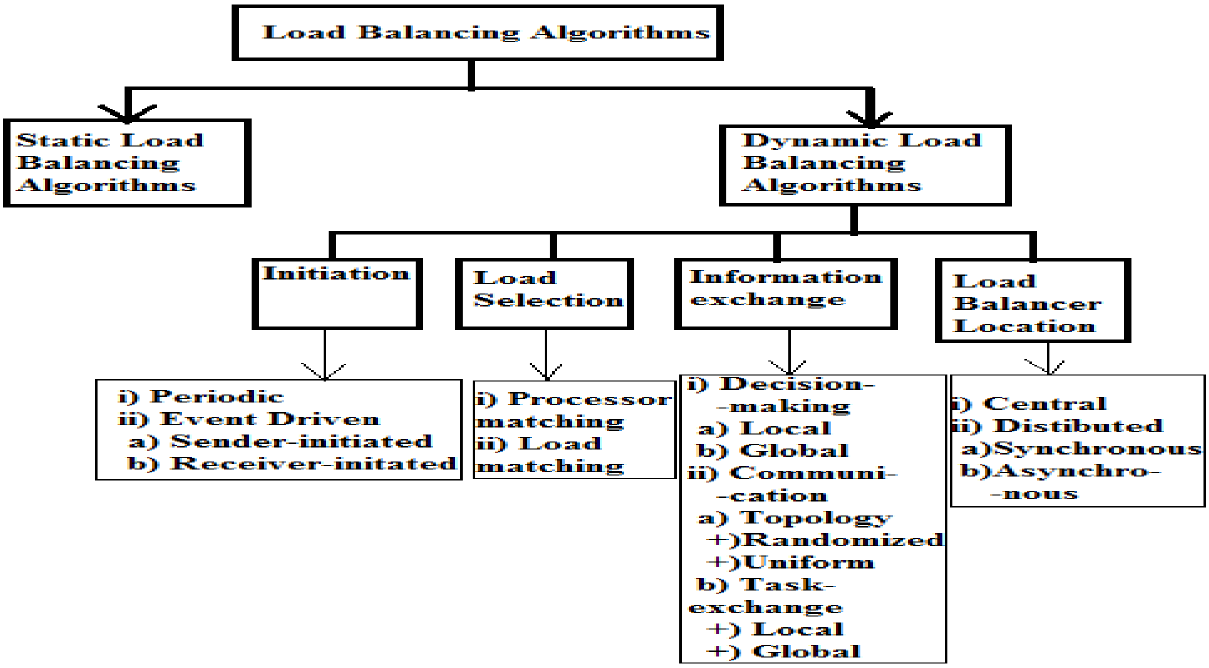


Fig 2: Classification of load Balancing Algorithms[2][4]

Existing Load Balancing Algorithm cab be summarized by using following table:-

Table 1: Comparatively study of various load balancing algorithms

Algorithm	Static Environ- -ment	Dynamic Environ- -ment	Centralized Balancing	Distributed Balancing	Hierar- -chical Balancing	Pros	Cons
Round- Robin[2][3] [5][8]	Yes	No	Yes	No	No	Equal distribution of work load	Job processing time is not considered.
Weighted Round Robin[5][8]	Yes	No	No	Yes	No	Better utilization.	Processing time is not considered.
Randomized [2][3][8]	Yes	No	~	~	~	Works well when program is of equal load.	Does not maintaining deterministic approach.
Map Reduce [9][10]	Yes	No	No	Yes	Yes	Parallel execution of jobs. So, no overhead.	Complexity is more thereby low throughput is achieved.
Max- Min[2][5][8]	Yes	No	Yes	No	No	Reduce the make span.	Smaller job have to waitfor long time
Min- Min Algorithm [2][5][8]	Yes	No	Yes	No	No	Simplicity. Minimize the completion time of all works.	Does not consider the existing load on a resource. Thus, some nodes always get busy but some nodes maybe still idle.
OLB[2][5] [8][9] (opportunistics	Yes	No	Yes	No	No	Keeps each node in the	Does not consider the

load balancing)						cloud busy.	execution time of the task. Make span is very poor.
MET[5] (Minimum execution time)	Yes	No	Yes	No	No	Improves the makespan over OLB.	Causes a severe load imbalance.
MCT[5] (Minimum Completion Time algorithms)	Yes	No	Yes	No	No	Improves makespan over MET.	Consider only one job at a time.
CLBDM[8] [9][10] (Central Load Balancing Decision Model)	Yes	No	Yes	No	No	Session Switching. Suitable for heterogeneous environment.	Due to dependency on RR algorithm, if the connection goes above the threshold level problem arises.
LBMM [5][8][9][10] **(load balance min-** **min)**	Yes	No	No	No	Yes	Improves the load balancing of min-min. & minimize the execution time of each node.	Doesn’t state how to select a node for a complicated task requiring large-scale computation.
Dynamic Round Robin[8]	No	Yes	Yes	Yes	No	Reduce the power consumption cost.	Does not scale up for large data center.
ESCE[5][8] **(Equally Spread Current Execution)**	No	Yes	Yes	No	No	Improve Response time and processing time of a job.	Not fault tolerant because of single point failure.
Biased Random Sampling [2][5][8][9]	No	Yes	No	Yes	No	Suitable for large n/w system.	Degrades when load increases.
Event Driven Algorithm **[9]**	No	Yes	No	Yes	~	Generates the game session load balancing algorithms.	Limited Application
Particle Swarm Opt.	No	Yes	No	Yes	No	Useful in large and heterogeneous environment	High complexity and overhead.
Honey Bee Foraging Algorithm [2][5][6][8] **[9]**	No	Yes	No	Yes	Yes	Achieves global load balancing through local server action. Maximize the throughput. Minimum waiting time and overhead.	Throughput decreases when variety of nodes increases.
Genetic Algorithm	No	Yes	Yes	No	No	Useful in heterogeneous environment.	High complexity and time consuming. Not fault tolerant.
Ant Colony[2] **[9][10]**	No	Yes	No	Yes	No	Allow shared computation and storage over long distances. Achieves	Network overhead due to large number of ants.

						optimal job scheduling. Global optimization.	
Active Clustering **[2][5][8][9]**	No	Yes	No	Yes	No	Balances load efficiently.	Perform poorly in heterogeneous environment.
Throttled Load Balancer [5][8]	No	Yes	No	Yes	Yes	To distribute the load evenly among the virtual machines.	Does not consider the current load on virtual machines.
WLC (Weighted Least **Connection)[8]**	No	Yes	Yes	No	No	Balance load efficiently.	Processing speed & storage capacity aren't considered.
ESWLC(Exponential smoothing WLC)[8] **[10]**	No	Yes	Yes	No	No	Capabilities of each node are considered.	Complex calculations are involved.
INS(Index name server) **[10][12]**	No	Yes	No	Yes	~	Enhance performance, reduce duplicity and redundancy	High complexity. No forecasting algorithm used.

4. CLOUD TESTING

According to Wikipedia, "cloud testing is a form of software testing in which web applications that leverage cloud computing environments (cloud) seek to simulate real-world user traffic as a means of load testing and stress testing web sites. The ability and costs to simulate web traffic for software testing purposes has been an inhibitor to overall web reliability".

Various issues, challenges and needs of cloud testing is discussed in [15].There are different types of cloud testing available such as:

1. Functional Testing
2. Load Testing
3. Stress Testing
4. Latency Testing
5. Performance Testing
6. Browser Performance Testing
7. Compatibility Testing.

Considered parameters for performance testingare shown below:-

1. **Response Time: -** Total time taken per second that is used in responding to the request. If N is total number of responses, then average response time is calculated as:-

   Average response time $=\sum \frac{Response\ \ time\ (\text{sec})}{N}$

2. **Throughput: -**Ratio of total amount of data used for completion of tasks to the total time consumed in the experiment. High throughput is the desired requirement of the system. It is generally calculated in terms of Kilobytes per second.

   Throughput $= \frac{Amount\ \ data}{Time}$

3. **Overhead: -** It is calculation of nimiety expenses that are necessary for the ongoing process; this can include computation time, memory, bandwidth and other resources. Minimum overhead is the requirement of the system.
4. **Fault Tolerance: -** Fault tolerance is the characteristic of a system because of which it can operate properly even some of its components are damaged. In this we test our system on the different conditions.
5. **Migration Time: -** It is the total time in movement of required resources from one position to another. It includes data migration, live migration etc.
6. **Resource Utilization:-**It is the testing of system for utilization of resources efficiently so that we can maximize output, minimize response time.
7. **Scalability: -**It is the testing of application to measure its capabilities to scale up or scale down in terms of changing size(number of nodes) with time.
8. **Performance**:- It is testing of overall effectiveness of different characteristics of the system

5. RELATIONSHIP BETWEEN LOAD BALANCING ALGORITHMS AND PARAMETERS OF PERFORMANCE TESTING

Here, a logical view is presented between different load balancing algorithms and the characteristics which impact the outcome of these algorithms and also testing of the algorithms.

Table2. Load Balancing Techniques v/s Parameters of Performance Testing

Metrics/ Techniques	Overhead	Fault tolerance	Migration time	Response time	Resource Utilization	Scalability	Performance
Round Robin[2][3][5][8]	Yes	No	No	Yes	Yes	Yes	Yes
Dynamic Round Robin [8]	Yes	Yes	Yes	No	Yes	No	No
PALB (Power aware load balancing)	Yes	Yes	Yes	Yes	Yes	No	No
Active Monitoring (AMLB)[8]	Yes	No	Yes	Yes	Yes	Yes	No
FAMLB	Yes	Yes	Yes	No	Yes	Yes	Yes
Min-Min [2][5][8]	Yes	No	No	Yes	Yes	No	Yes
Max-Min [2][5][8]	Yes	No	No	Yes	Yes	No	Yes
OLB+LBM [2][5][8][9]	No	No	No	No	Yes	No	Yes
Throttled [5][8]	No	Yes	Yes	Yes	Yes	Yes	Yes
Honeybee Foraging [2][5][6][8][9][11]	No	No	No	No	Yes	No	No
Active Clustering [2][5][8][9][11]	Yes	No	Yes	No	Yes	No	No
Biased Random Sampling [2][5][8][9][11]	Yes	No	No	No	Yes	No	Yes
Ant Colony	Yes	Yes	~	No	Yes	Yes	Yes
Map Reduce	Yes	Yes	~	~	~	~	~
INS(Index Name Server)[10][12]	Yes	No	Yes	Yes	~	~	~

6. CONCLUSION

In this study, a number of load balancing algorithms are considered; their positive and negative points are studied as well as their methodology. We also study about cloud testing, different types of cloud testing and their parameter. Study initially starts from round robin technique to the latest index name server technique with a generalized point of view like the field in which they are used and whether they are static, dynamic, centralized, decentralized and hierarchical. Table 2 provides relationship between various parameters and load balancing algorithms. Using these results, mathematical equations can be derived for various load balancing algorithms to carry out mathematical modulation. Mathematical modulation helps in development of systems or redefining models and better understanding of system. Further, repetitive analysis can be done of the process for better results and model can be redefined.

7. REFERENCES

[1] T. Cláudio, S.P. Joaquim, A. Ricardo, B. Tiago and M. André, "The building blocks of a PaaS", J NetwSyst Manage, Springer, 2014, pp.75-99.

[2] K. Rajwinder and L. Pawan, "Load balancing in cloud computing", in Proceedings of Int. Conf. on Recent Trends in information, Telecommunication and Computing, ITC, ACEEE, 2014, pp.374-381.

[3] R. Soumy and D.S. Ajanta, "Execution analysis of load balancing algorithms in cloud computing environment", International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.5, October 2012, pp. 1-13.

[4] K. Zubair, S. Ravendra, A. Jahangir and S. Shailesh, "Classification of Load Balancing Conditions for parallel and distributed systems", International Journal of Computer Science Issues(IJCSI), Vol. 8, Issue 5, No 1, September 2011, pp. 411-419.

[5] S. M. Shanti, D. R. Rajadeepan and P. Digamber, "Analysis of Load Balancers in Cloud Computing", International Journal of Computer Science and Engineering(IJCSE), Vol. 2, Issue 2, 2013,pp. 101-108.

[6] G. R. Rajesh and V. Jeyakrishnan, "A Survey on Load Balancing in Cloud Computing Environments", International Journal of Advanced Research in Computer and Communication Engineering(IJARCCE), Vol. 2, Issue 12, 2013, pp. 4726-4728.

[7] R. Raju, R. G. Babukarthik, D. Chandramoha, P. Dhavachelvan and T. Vengattaraman, "Minimizing the Makespan using Hybrid Algorithm for Cloud Computing", IEEE, 2012.

[8] B. S. Subhadra and K. S. Asbhadoria, "A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment", 5th International Conference on Computer and Communication Technology(ICCCT), IEEE, 2014, pp. 87-95.

[9] J. Uma, V. Ramasamy, A. Kaleeswaran, "Load Balancing Algorithms in Cloud Computing Environment – A Methodical Comparison", International Journal of Advanced Research in Computer Engineering & Technology(IJARCET), Volume 3, Issue 2, 2014, pp.272-275.

[10] A. N. Klaithem, M. Nader, A. N. Mariam and A. J. Jameela, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", Second Symposium on Network Cloud Computing Applications, IEEE, 2012, pp. 137-142.

[11] R. Martin, L. David, A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 24th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2010, pp. 551- 556.

[12] Y. W. Tin, T. L. Wei, S. L. Yu, S. L. Yih, L. C. Hung, S. H. Jhih, "Dynamic Load Balancing Mechanism based on Cloud Storage", IEEE, 2012, pp. 102-106.

[13] J. Wang and M. Fenpeng, "Software Testing Based on Cloud Computing", IEEE International Conference on Internet Computing and Information Services, 2011, pp. 176-178.

[14] C.M. Geetha and K.G. Mohan, "Two Level Hierarchical Models of Load Balancing in Cloud", International Journal of Emerging Technology and Advanced Engineering, 2013, pp. 307-311.

[15] G. Jerry, Y.B. Xiaoying and T. Wei-Tek, "Cloud Testing- Issues, Challenges, Needs and Practice", Software Engineering: An International Journal (SEIJ), Vol. 1, No. 1, 2011, pp. 9-23.

[16] Y.B. Xiaoying, Muyang Li, C. Bin, Z.T. Wei-Tek and G. Jerry, "Cloud Testing Tools", Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering (SOSE 2011), p. 1-12.

[17] B. Ekkarat, "Performance and Security Issue on Open Source Private Cloud", IEEE, Proceedings of the International Electrical Engineering Congress, 2014.